10.1 Advice for applying Machine Learning

Debugging a learning algorithm:

- Suppose you have implemented regularized linear to predict housing pricing

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{m}\theta_j^2\right]$$

However, when you test your hypothesis on a new set of houses, you find that it makes unacceptably large errors in its predictions. What should you try next?

Get more training example → fixed high variance
Try smaller set of features → fixed high variance
Try getting additional features → fixes high bias
Try adding polynomial features → fixes high bias
Try decreasing λ → fixes high bias
Try increasing λ → fixes high variance

Machine Learning diagnostic.

Diagnostic: A test that you can run to gain insight what is/isn't working with a learning algorithm, and gain guidance as to how best to improve performance

Diagnostic can take time to implement, but doing so can be very good use of time

10.2 Evaluating your hypothesis

Training / testing procedure for linear regression

- Learn parameter θ from training data (minimizing training error $J(\theta)$)
- Compute test set error:

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

- Misclassification error (0/1 misclassification error):

$$err(h_\theta(x), y) = \begin{cases} 1 & \text{if } h_\theta \geq 0.5, \ y = 0 \\ 0 & \text{if } h_\theta < 0.5 \ \ y = 1 \end{cases} \Big\} \text{ error}$$

$$\text{Test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{error}} err(h_\theta(x_{test}^{(i)}), y_{test}^{(i)})$$

10.3 Model Selection and Train Validation

1. $h_\theta(x) = \theta_0 + \theta_1 x$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

⋮

10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

Choose $\theta_0 + \cdots \theta_5 x^5$

How well does the model generalise? Report test set error $J_{test}(\theta^{(5)})$

Problem: $J_{test}(\theta^{(5)})$ is likely to be an optimistic estimate of generalisation error. i.e our extra parameter (d = degree of polynomial) is fit to test set.

Evaluating the hypotheses

60% - training set

20% - cross validation set

20% - test set

Train / Validation / test error

Training error:

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Cross validation error:

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$$

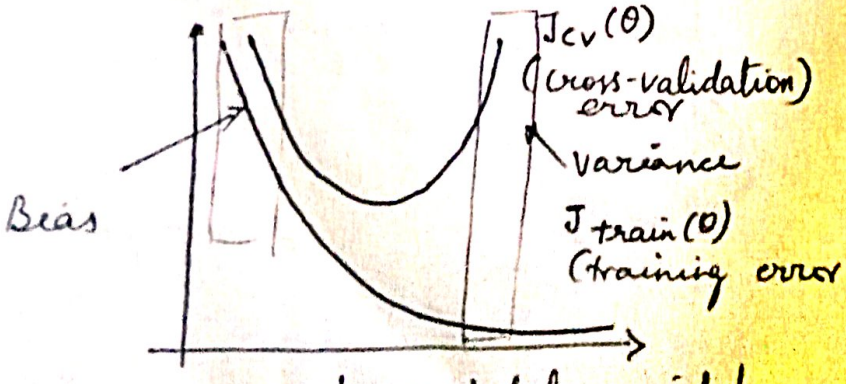
Test error:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m}\left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)}\right)^2$$

- Select the model which give least validation error
- Estimate the generalisation error for the model

10.4 : Diagnosing Bias Vs Variance

Suppose your learning algorithm is performing less well than you were hoping ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high). Is it bias problem or a variance problem?



Bias (underfit)
- $J_{train}(\theta)$ will be high
- $J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit)
- $J_{train}(\theta)$ will be low
- $J_{cv}(\theta) >> J_{train}(\theta)$

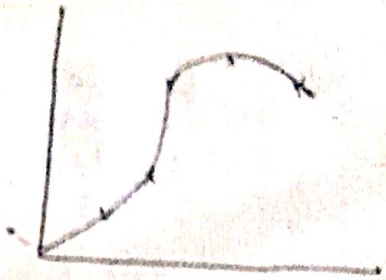
10.5 Linear regression with regularisation

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

Large λ
High bias (underfit)
$\lambda = 1000$, $\theta_1 \approx 0$, $\theta_2 \approx 0$ ...

Intermediate (λ)
Just right



Small (λ)
High variance (overfit)

Choosing the regularization parameter λ

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{i=1}^{m}\theta_j^2$$

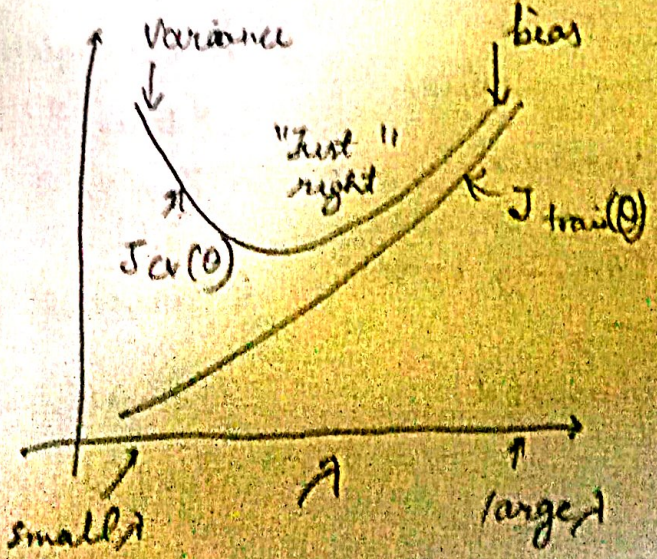
Try $\lambda = 0 \longrightarrow \min_\theta J(\theta) \rightarrow \theta^{(1)} \longrightarrow J_{cv}(\theta^{(1)})$

$\lambda = 0.01 \longrightarrow \min_\theta J(\theta) \rightarrow \theta^{(2)} \longrightarrow J_{cv}(\theta^{(2)})$

$\lambda = 0.02$

$\lambda = 0.08 \longrightarrow \theta^{(12)} \longrightarrow J_{cv}(\theta^{(12)})$

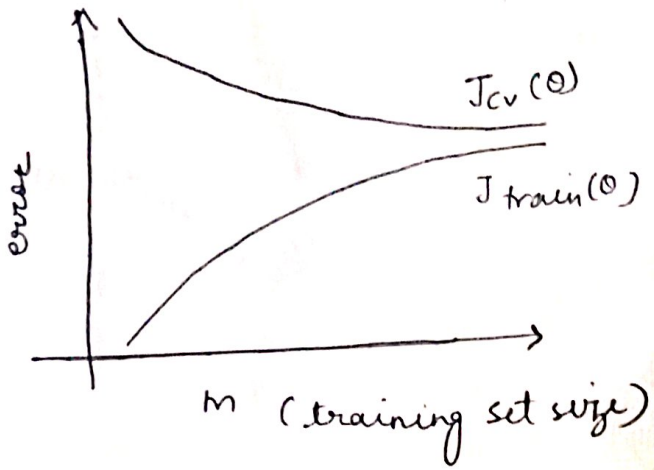
Pick which give least cross validation error



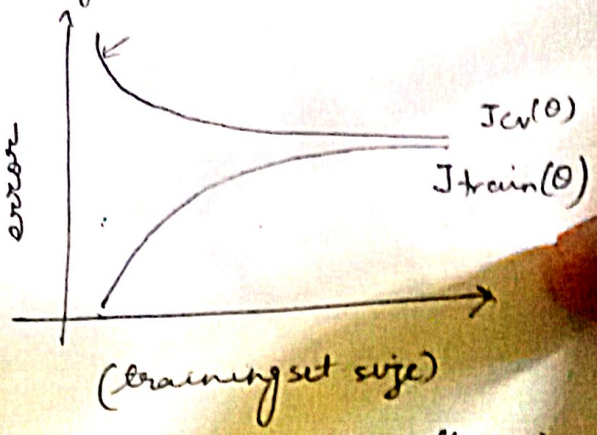
10.6 Learning Curves

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2$$

$$J_{cv}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2$$

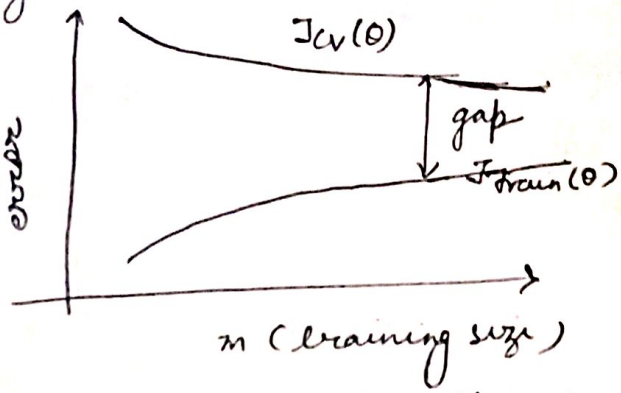


High bias



If learning algorithm is suffering from high bias, getting more training data will not (by itself) help much

High Variance



If a learning algorithm is suffering from high variance, getting more training data is likely to help

11 | Machine Learning System Design | Prioritising what to work on

Building a spam classifier

- Supervised learning. x = features of email, y = spam(1) or spam(0). Features x: Choose 100 words indicative of spam / not spam

Note In practise, take most frequently occurring n words (10,000 to 50,000) in training set, rather than manually pick 100 words

How to spend you time to make it have less error?

- Collect lots of data
  - Eg. "honeypot" project
- Develop sophosticated features for message body, eg should "discount" and "discounts" be treated as the same word? How about "deal" and "dealer"? Features about punctuation?
- Develop sophosticated algorithm to detect misspelling (e.g. mortgage, medicine, w4ches).

12 | Error Analysis

m_{cv} = 500 examples in cross validation set
Algorithm misclassifies 100 emails
Manually examine the 100 errors, and categorize them based on:

i) What type of email it is
ii) What cues (features) you think would have helped the algorithm classify them correctly.

Pharma
Replica / Fake
Steal passwords
Other

→ Deliberate misspellings (m0rgage, medicine, etc)
→ Unusual email routing
→ Unusual (spamming) punctuation

The importance of numerical evaluation

Should discount / discounts / discounted / discounting be treated as the same word?

Can use "streaming" software (E.g "Porter stemmer")

universe / university

Error analysis may not be helpful for deciding if this is likely to improve performance. Only solution is to try it and see if it works

Need numerical evaluation (e.g. cross validation error) of algorithm's performance with and without stemming.

Without stemming: with stemming:

Distinguish upper vs. lower case (Mom / mom)

11.3. Error Metrics for Skewed Classes

Cancer classification example

Train logistic regression model $h_\theta(x)$. ($y=1$ if cancer, $y=0$ otherwise)

Find that you got 1% error on test set.

(99% correct diagonoses)

Only 0.50% of patients have cancer

↳ skewed classes

Precision / Recall

$y=1$ in presence of rare class that we want to detect

Actual Values

Predicted class	1	0
1	True Positives	False Positives
0	False negatives	True negative

Precision

(Of all patient where we predicted $y=1$, what fraction has cancer?)

$$\frac{\text{True positives}}{\#\text{predicted positives}} = \frac{\text{True positive}}{\text{True pos + Fake pos}}$$

Recall

(Of all patients that actually have cancer what fraction did we correctly detect as having cancer)

$$\frac{\text{True Positives}}{\#\text{ actual positive}} = \frac{\text{True positive}}{\text{True pos + Fal neg}}$$

11.4 Trade off precision and recall

Logistic regression: $0 \leq h_\theta(x) \leq 1$

Predicte 1 if $h_\theta(x) \geq 0.5$

Predict 0 if $h_\theta(x) < 0.5$

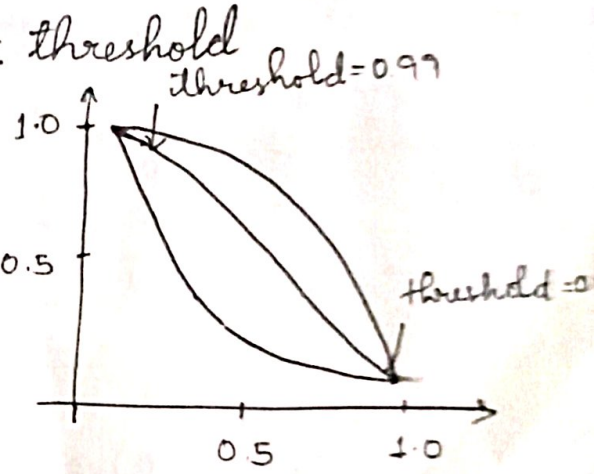
Suppose we want to predict $y=1$ (cancer) only if very confident

→ Higher precision, lower recall

Suppose we want to avoid missing too many cases of cancer (avoid false negatives).

→ Higher recall, lower precision

More generally. Predict 1 if $h_\theta(x) \geq$ threshold



F Score (F score)

How to compare precision/recall numbers?

	Precision (P)	Recall (R)
Algorithm 1	0.5	0.4
Algorithm 2	0.7	0.1
Algorithm 3	0.02	1.0

Average $\frac{P+R}{2}$ ✗

F1 score $\frac{2PR}{P+R}$

11.5. System Design

Designing a high accuracy learning system

E.g. Classify between confusable words

{to, two, too} {then, than}

→ for breakfast I ate — eggs.

Algorithms
- Perceptron (Logistic regression)
- Winnow
- Memory-based
- Naive Bayes

Large Data Rationale

Assume features $x \in \mathbb{R}^{n+1}$ has sufficiently information to predict y accurately

Example: For a breakfast I ate ____ eggs

Counterexample: Predict housing price from only size ($feet^2$) and no other features

Useful test: Given the input x, can a human expert confidently predict y?

→ Use a learning algorithm with many parameters (e.g. logistic regression / linear regression with many features; neural network with many hidden units).

low bias algorithm

→ $J_{train}(\theta)$ will be small

Use a very large training set (unlikely to overfit) ← low variance

→ $J_{train}(\theta) \approx J_{test}(\theta)$

$J_{test}(\theta)$ will be small